Qubo III: Slow and Steady

Josh Smith
Software Lead
Robotics at Maryland
College Park, MD
jms44@terpmail.umd.edu

Nathaniel Renegar
Embedded Lead
Robotics at Maryland
College Park, MD
nrenegar@umd.edu

Rainier Hood
Soldering Lead
Robotics at Maryland
Cambridge, MD
rainierhood@gmail.com

Dillon Capilongo
Co-Mechanical Lead
Robotics at Maryland
College Park, MD
cmdjcap2@comcast.net

Mustafa Khan
Co-Mechanical Lead
Robotics at Maryland
College Park, MD
muskha29@gmail.com

***Abstract*—For the 2022 competition, Robotics at Maryland will submit an improved version of Qubo. This version of Qubo has had a significant software refactor, but maintains the same hardware from previous competitions. Qubo will attempt a subset of the competition tasks, relying on a suite of sensors to make sense of the underwater environment. This design shows good competition performance can be obtained without the newest hardware.**

***Index Terms*—Autonomous underwater vehicle (AUV), Printed circuit board (PCB)**

I. Introduction

Qubo is an autonomous underwater vehicle (AUV) that has been continuously improved since 2016. This year, Qubo's aging electrical and software systems have been revised. A clean ROS node structure was implemented, which allowed higher level autonomy to be abstracted from whether the robot was running in simulation or in the real pool. This allowed us to build off of our remote work throughout the pandemic, not having to re-implement all of our code now that we are competing with a physical robot.

II. Competition Strategy

This year's strategy was to accomplish as much possible with minimum modifications of Qubo. Maximizing the capability of existing hardware is both low cost and identifies areas where the design may be improved. Therefore, our compute hardware, sensors, vehicle frame, hull, etc. are largely the same from the 2018 competition (See Appendix).

We aim to achieve a subset of the tasks outlined in the Team Manual. First, Qubo will navigate through the gate and make attempts at all bonus points (coin flip, fixed heading, and style). After proceeding through the gate, Qubo will navigate to the buoys, touching the buoy corresponding to which side of the gate was traversed. After this, Qubo will navigate to the bins. Using a camera, velocity sensors, and inertial sensors, Qubo will loop a rigid hook onto the handle of the bin and slide it open. Markers will then be dropped into the bin. After this segment of the run, Qubo will then use a multilateration navigation system to find the hexagon task. At this point, Qubo will home in onto the pinger, then breach.

To achieve this planned run, Qubo features a number of sensors and actuators.

III. Design Creativity

A. Mechanical

We have designed a torpedo launcher based off a self-propelled torpedo pool toy. This allows us to launch the torpedo with less force, and allows the torpedo to maintain its velocity. We also have designed a passive marker dropping system that allows us to drop a marker simply by inverting the robot along an axis. These methods allow us to simplify the mechanisms and allow for fewer and less complex actuators.

B. Software

We have refactored our codebase to create an abstraction so that the majority of our code does not change, whether we are running in our Gazebo simulation, or running on our physical robot. Our sensor read nodes on the physical robot use the same interface as the simulated sensors in our simulation.

IV. Experimental Results

A. Electrical System

The electrical system features three boards that regulate power and route signals between the various processors, sensors and actuators on the robot. The individual components are listed below. The first three boards sit in the main electronics hull, while the battery support board sits inside the dedicated battery hulls.

- ARM Processor Board
- Power Supply Board
- Batteries
- Thrusters
- Battery Support Board

This year, several changes were made to the electrical system that improve the functioning of the robot as a whole. An I2C signal is sent by the TiVa microcontroller to the interface PCB which has a chip that decodes the I2C into 8 separate PWM signals (one per thruster ESC). Thus the Hypertronics system acts as the primary physical interface between the thrusters and the main CPU. The addition of this connector is also what allowed for the entire electrical subsystem (three PCB's and 2 CPUs) to be mounted on rails that slide in and out of the electrical hull.

A battery board was designed to sit on the battery hulls and act as an ideal diode. Each battery board prevents reverse current from entering the battery it is connected to. This prevents batteries from charging each other. The battery boards also shut off power from the batteries to the rest of the robot if they detect user-selected over-voltage, under-voltage, or over-current scenarios. By adding these boards the robot has become safer to operate since the likelihood of a battery-related accident is now lower.

B. Mechanical System

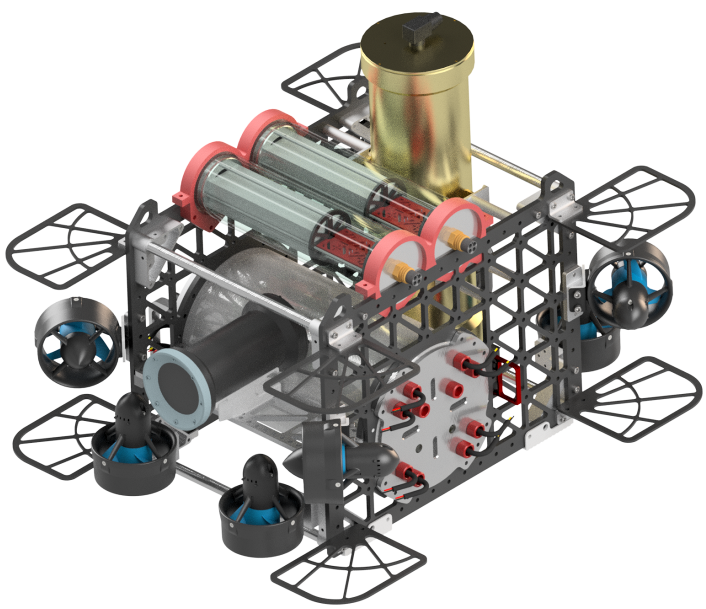

Fig. 1. CAD Render of Qubo, with preliminary DVL mounting

Qubo's mechanical system is composed of a main pressure hull, eight Blue Robotics T200 thrusters, a camera hulls, and an aluminum frame. The main pressure hull houses the electronics system and the computers. The robot relies heavily on vision for maneuvering. Mechanisms for firing a torpedo, dropping markers, and manipulating objects are currently being manufactured and implemented.

The mechanical design of Qubo emphasizes modularity and ease of machining. One of the lessons learned from Qubo's predecessor, Tortuga IV, was that a simple frame design can reduce the serviceability of and limit access to internal components. In Qubo, the frame was designed with respect to the hardware configuration. The design eliminates the spatial dependency of systems, and allows for quick part removal, swapping, and modification. Additionally, the robot was designed to reduce machining complexity. Only a few parts require CNC milling, while others can be manually machined or 3D printed.

The propulsion system is intended to give the robot fine control of its movements. In RoboSub, the ability to perform forward, strafing, and yaw motion while maintaining depth is critical. On Qubo, four downward facing thrusters are placed in a rectangular pattern. Together, they maintain a horizontal operating plane which other four thrusters move on. The sideways thrusters are aligned to provide vector thrust 45 degrees from the center plane. Ideally, they provide more thrust for forward motion, and direct control for yaw. Further, the high degree of freedom allows the robot to perform complex maneuvers if so desired. This configuration requires the thruster to quickly reciprocate and deliver similar forward and backward thrust. Each thruster is mounted via an adapter that can be quickly detached and swapped in case of a malfunction.

1) Frame: The frame consists of two water-jetted aluminum panels with four main aluminum rods holding them together. The water jetted pattern offers option to mount additional hardware. The front and back plate provide additional mounting space. Each side is reinforced with cross beams. Stress analysis shows the frame can withstand several times its operational load. To protect the thrusters in the event of a collision, laser-cut Delrin bumpers are fitted onto the frame. To prevent corrosion, all fasteners used are stainless steel and the aluminum members are protected by a sacrificial anode.

2) Battery Hull: The two batteries are housed in separate, cast acrylic battery hulls sealed with aluminum endcaps on either side. The endcaps were fabricated from stock 6061 aluminum rods by manual lathe and mill machining. The battery hulls are mounted inside of the top of the frame so that they can be removed easily, and are mechanically constrained by the frame. This helps ensure that the hull remains sealed as the battery heats up and the internal pressure of the hull changes. The battery and battery boards are mounted upon 4 orthogonal stainless steel hexagonal stay rods screwed into the inside of the endcaps. All of the support mounting for the hull and the battery boards are made using 3D printed PLA.

C. Software

The software system for Qubo is divided into two categories depending on which computer they run on. High level software runs on the Jetson, and embedded software runs on the Tiva. Most of the code base for Qubo has remained the same as previous years, maintain the same system levle organization and the reliance on ROS.

1) High Level Software: The high level software system on Qubo has remained relatively untouched since Robotics at Maryland's last competition. It was written using the Robotic Operating Sustem (ROS) as a base, to take advantage of the large number of utilities available and community support. Work was done on the control system after the last competition, using the simulation software Gazebo and the Gazebo plugin UUVSimulatior to simulate underwater environments. Members worked to perfect and experiment with the control system presented in the last report, a simple PID control along each degree of freedom. Using the simulator and a simplified model of the robot, base values for all PID constants were found.

2) Vision Software: Work was also done on an OpenCV based vision system to recognize orange gates for the pre-qual task. Using the known dimensions of the gate, once the gate is detected the relative global position and orientation of the gate can be estimated. Results were noisy, but this could be alleviated in the future with a simple filtering technique.

3) Embedded System: The embedded system saw the most work between competitions, as many problems were found during and before Robotics at Maryland's last competition. The system still uses the same general layout and design as the previous report; a TI Tiva C running various task using the FreeRTOS real time operating system. New tasks for the system were written to interact with the hardware in our system. Specifically, tasks to communicate with our I^2C based PWM controller, ADS chip, and temperature sensor were written and tested. Inter-task communication was also rewritten, and now uses FreeRTOS message buffers to transmit small amounts of data around the system. The embedded system still communicates with the high level system using Qubobus, a bytestream protocol developed by the team and described in the previous report.

V. Experimental Results

Due to issues during electrical and mechanical manufacturing, the team wasn't able to run the full system underwater before the competition, so all testing was done on land or in simulation.

A. Software

The new components of Qubo's software system was tested through multiple different methods, primarily using Gazebo and hardware testing. As most of the software was written before the hardware platform was fully assembled, individual components of the systems have not been tested together.

High level control software was tested using the ROS simulation software Gazebo. A basic computer model of the robot was created and exported to use in Gazebo, where basic control simulations were done using the UUVSimulator underwater Gazebo extension. While the model does have an accurate placement of the thrusters, and an estimated location for center of gravity, it does not have any information about the drag characteristics of the robot, so only simple control programming was done.

Sensors connected directly to Qubo's main computer were tested on land by running the relevant code, and verifying sensor values were correct.

1) Embedded: As the most complex software subsystem of the robot, Qubo's embedded system was tested more extensively than the others. Each task the system is responsible for was tested independently, with codependent tasks later tested together. Software that interfaced with sensors through data channels on the embedded computer was first tested using a development board containing all of the sensors, and then with the final electrical boards as they were manufactured. Sensor inputs and outputs were tested by connected oscilloscopes and signal generators to the pins of interested.

B. Electrical

The electrical subsystem was tested on a board by board basis. The battery balancing and power boards were tested with a power supply to ensure output voltages were correct with respect to the input voltages. Specifically, the over-voltage and over-draining features of the battery boards were verified. The regulated output voltages for the power board were also verified.

Acknowledgment

Robotics at Maryland would like to thank the Space Systems Laboratory at the Neutral Buoyancy Research Facility for lending their lab, time, and tools to the team. The team would also like to thank Dr. Akin specifically for acting as our team advisor over the years. Robotics at Maryland also thanks the Aerospace, Electrical and Computer Engineering, Mechanical Engineering and Computer Science departments for lending their space, time, and money to the team. Additionally, we thank Robotic Research LLC and the Maryland Robotics Center for their generous donations.

References

[1] S. Gillen et al., "Qubo: The Tortise Who Lived", 2017. [Online]. Available: https://www.robonation.org/sites/default/files/RS17_UMaryland_Paper.pdf

Appendix B: Component Specifications

Component	Vendor	Model/Type	Specs	Custom / Purchased	Cost	Year of Purchase
Buoyancy Control	NBRF Stock	Foam	Purple	Custom	0	N/A
Frame	Custom		Aluminum, Water Jetted	Custom	200	2017
Waterproof Housing	Blue Robotics	6in	Acrylic and Aluminum Endcaps	Purchased	400	2022
Waterproof Connectors	Blue Robotics	Penetrators	M10 Potted Connectors	Purchased	5	2022
Thrusters	Blue Robotics	T200	11.2 lbf forward thrust, 350 watt	Purchased	200	2017
Motor Control	Blue Robotics	Basic ESC	7-26 V, 30 amps max	Purchased	$36 x 8	2017
High Level Control	Texas Instruments	TM4C123GH6PM Dev Board	Arm Cortex-M4F	Purchased	20	2017
Actuators						
Propellers						
Battery	Gens Ace	GA-B-45C-5000-4S1P-Deans	14.8v, 5000mah	Purchased	36	2017
Converter	Custom PCB		12V, 5V, Fuse, Current and Voltage Monitoring	Custom	50	2017
CPU	Nvidia	Jetson TX1	ARM A57 and Maxwell GPU	Purchased	650	2017
Internal Comm Network	UART, GigE					
External Comm Network	Ethernet					
AHRS	PNI	Trax 1	2° accuracy heading and tilt	Purchased	1,000	2017
DVL	Teledyne	Explorer	+/- 5 m/s range, +/- 0.4% accuracy	Purchased	?	2006
Vision	Allied Vision	Mako G-131C	1280 x 1024 GigE Camera	Purchased	450	2017
Algorithms: Vision	OpenCV	Various basic vision processing algorithms			N/A	N/A
Algorithms: Other	Kalman Filter	State estimation and sensor fusion			N/A	N/A
Programming Language 1	C++ 14			N/A		
Programming Language 2	C			N/A		
Programming Language 3	Python			N/A		